SNS COLLEGE OF TECHNOLOGY

Coimbatore-35.

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with ‘A++’ Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

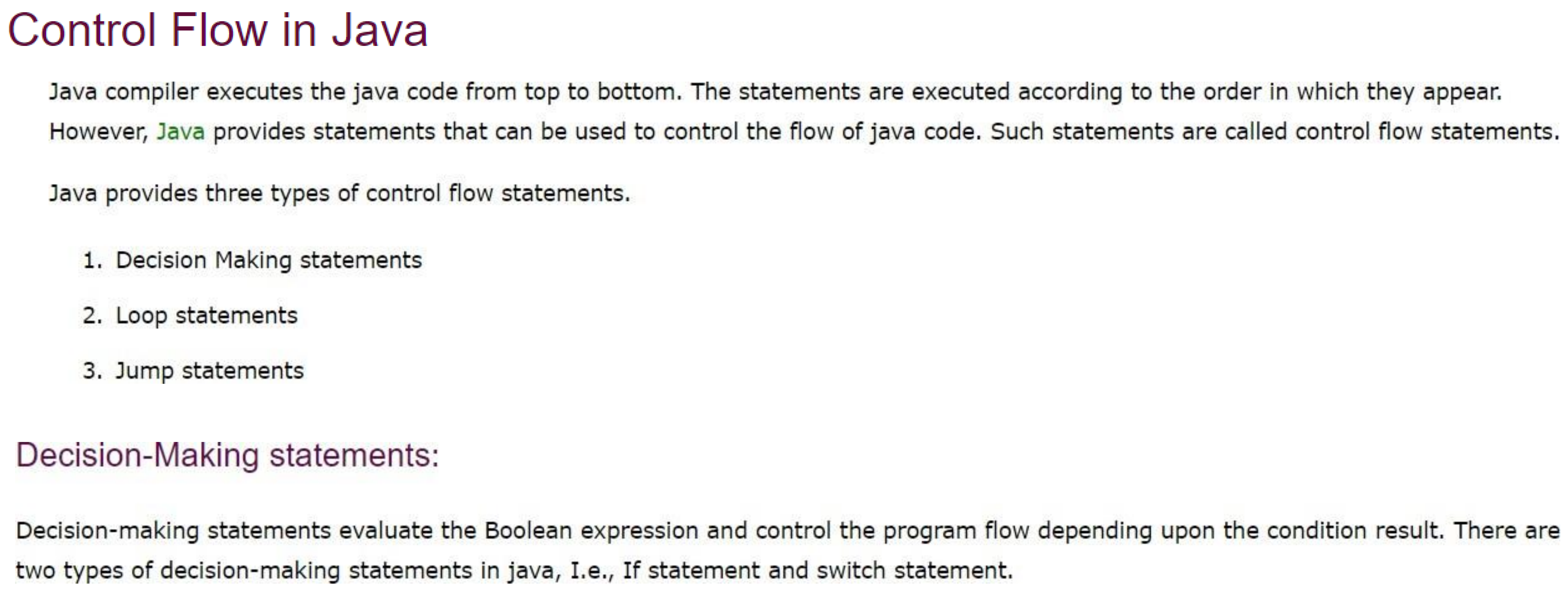
COURSE CODE & NAME : 23CST205 - Object Oriented Programming Using Java

II YEAR/ III SEMESTER

UNIT – II INTRODUCTION TO JAVA

Topic: BASICS OF JAVA PROGRAMMING-OPERATORS

Java Control Structures



Control Flow in Java

Java compiler executes the java code from top to bottom. The statements are executed according to the order in which they appear. However, Java provides statements that can be used to control the flow of java code. Such statements are called control flow statements.

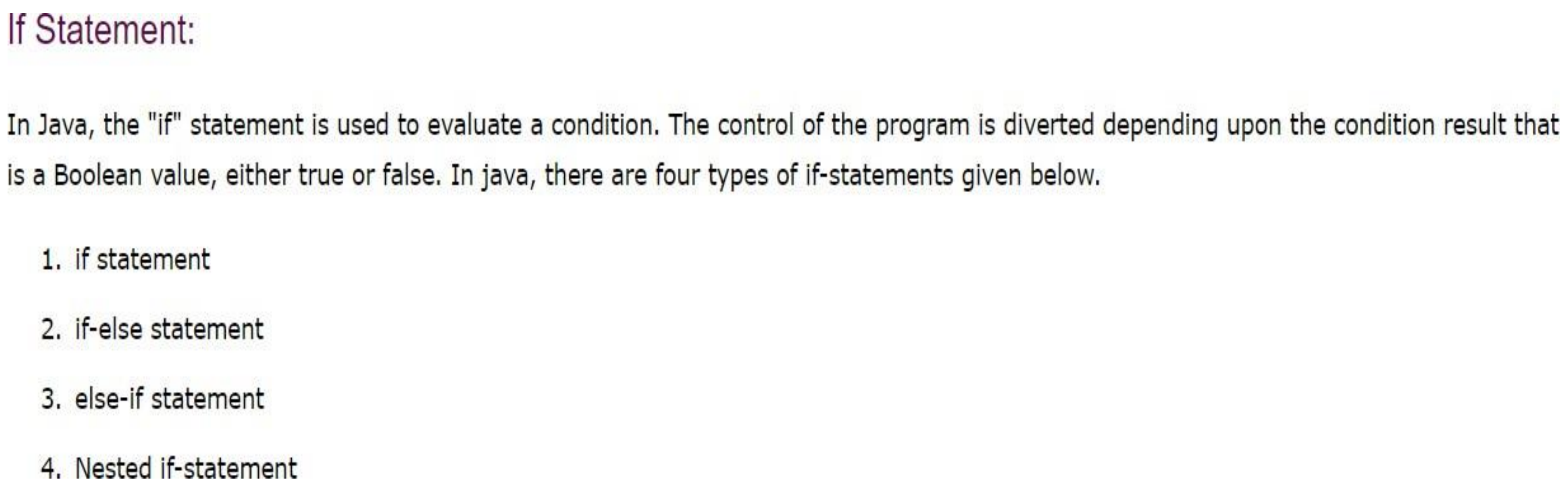
Java provides three types of control flow statements.

1. Decision Making statements
2. Loop statements
3. Jump statements

Decision-Making statements:

Decision-making statements evaluate the Boolean expression and control the program flow depending upon the condition result. There are two types of decision-making statements in java, I.e., If statement and switch statement.

if Statement

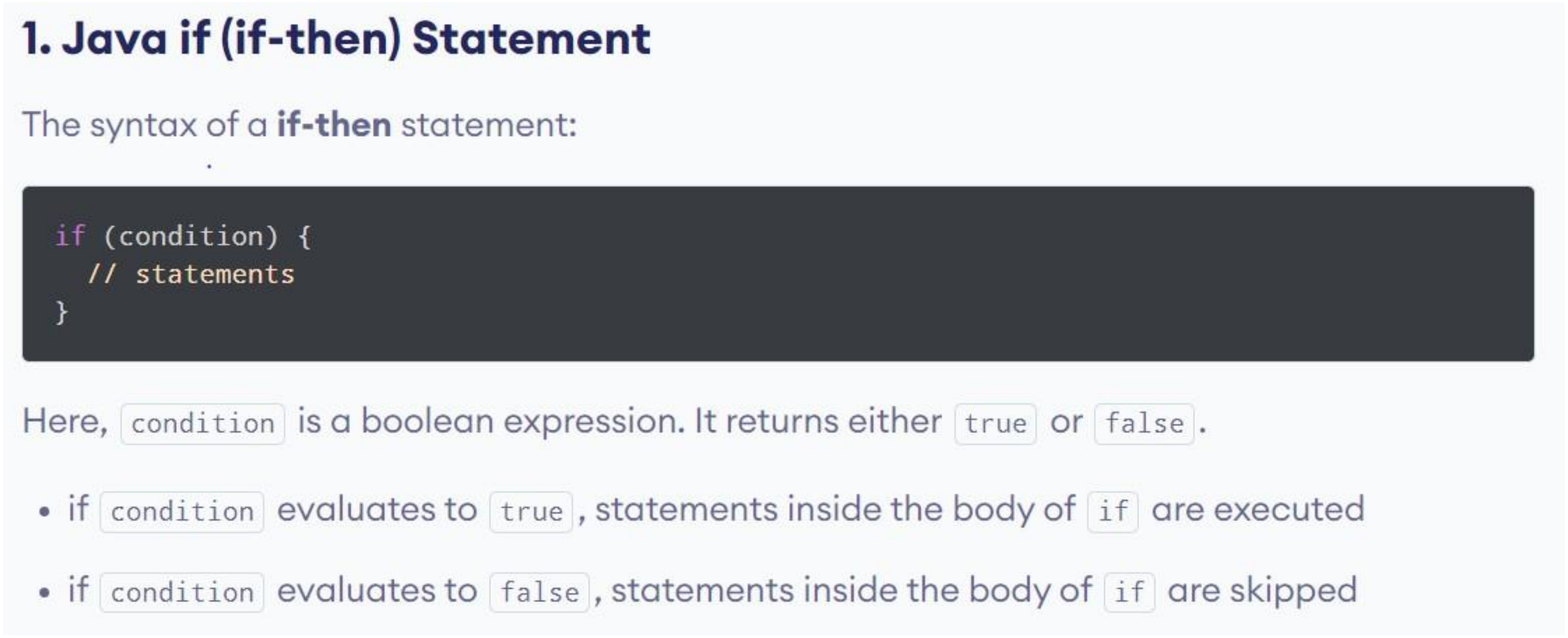


If Statement:

In Java, the "if" statement is used to evaluate a condition. The control of the program is diverted depending upon the condition result that is a Boolean value, either true or false. In java, there are four types of if-statements given below.

1. if statement
2. if-else statement
3. else-if statement
4. Nested if-statement

if Statement



1. Java if (if-then) Statement

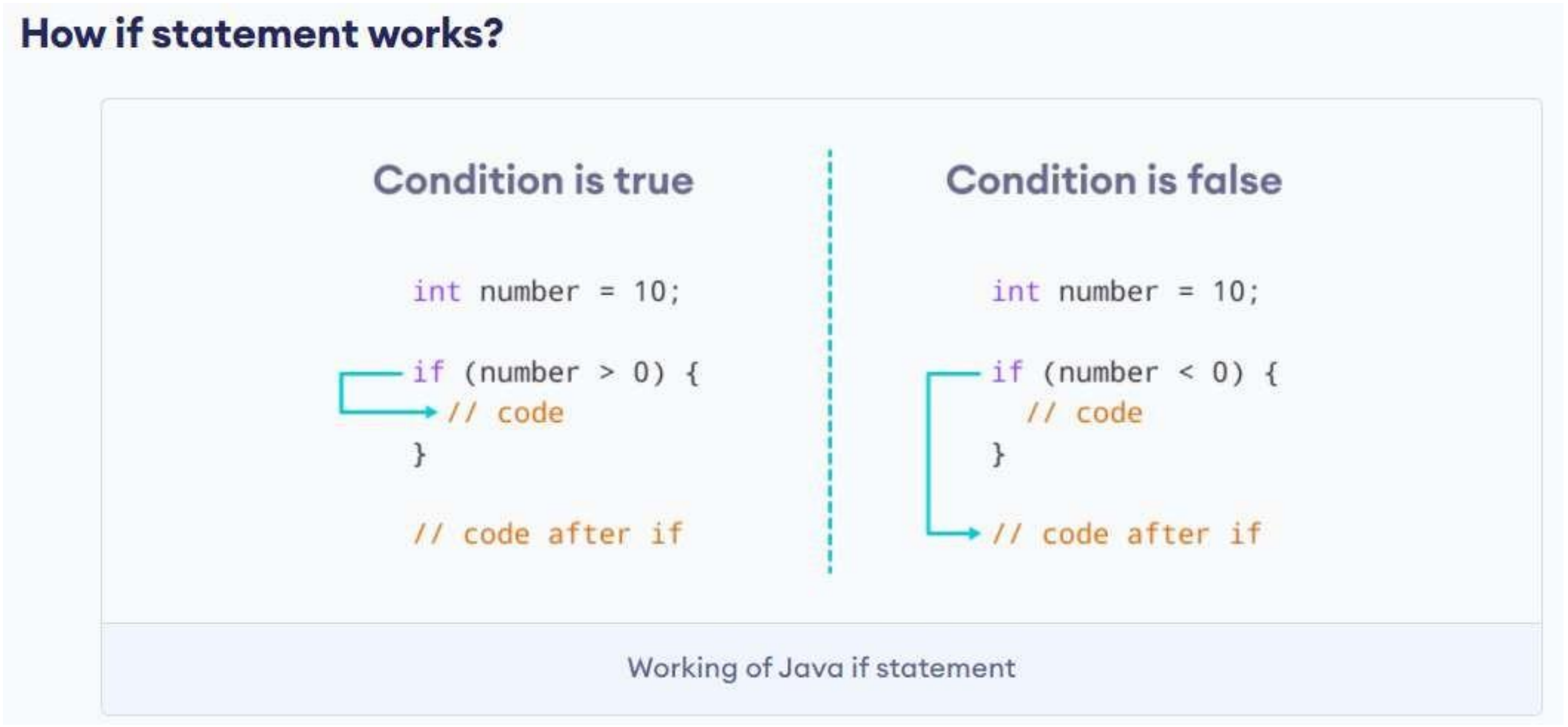
The syntax of a **if-then** statement:

```
if (condition) {
  // statements
}
```

Here, `condition` is a boolean expression. It returns either `true` or `false`.

- if `condition` evaluates to `true`, statements inside the body of `if` are executed
- if `condition` evaluates to `false`, statements inside the body of `if` are skipped

if Statement

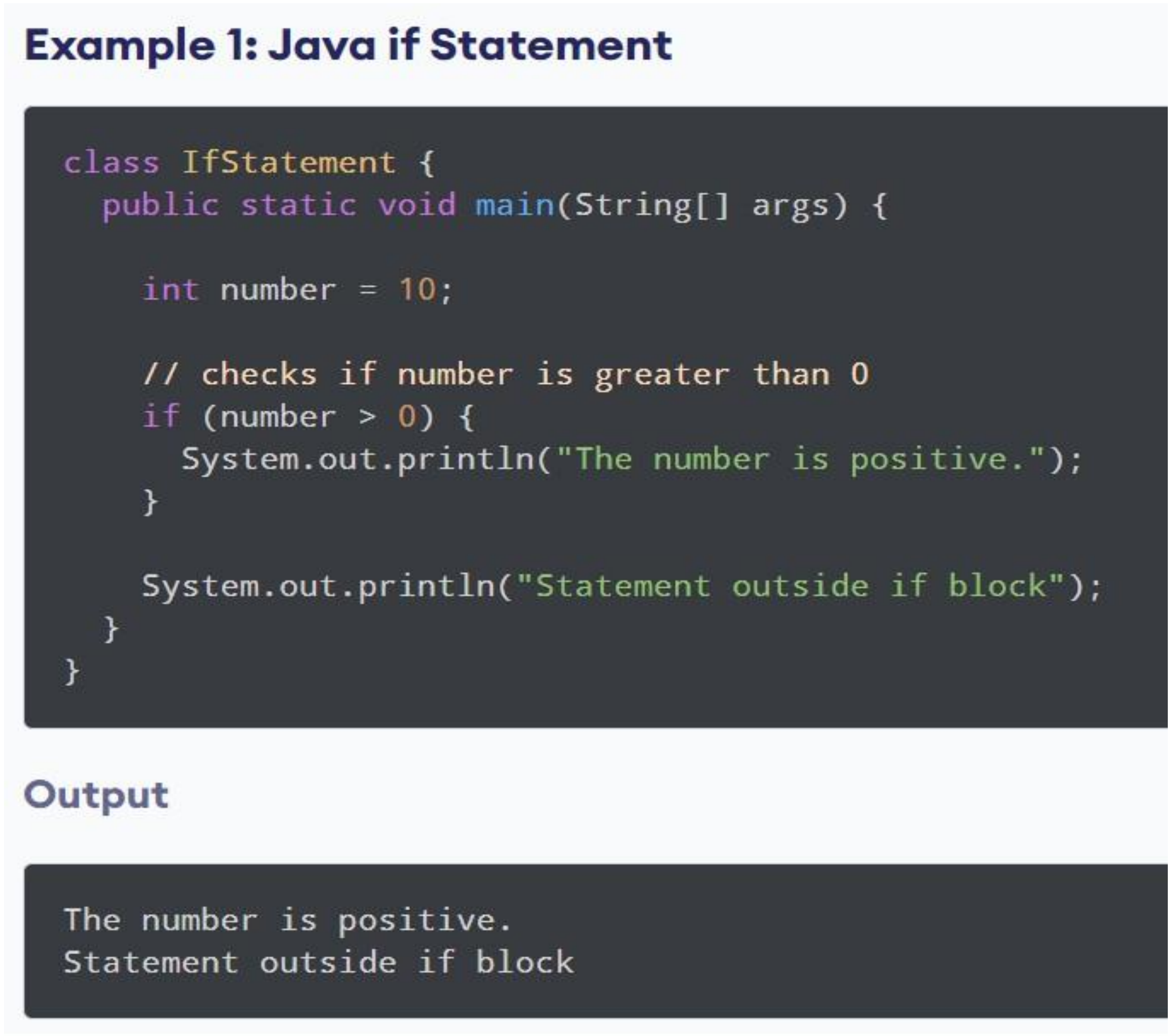


How if statement works?

```
int number = 10;

if (number > 0) {
  // code
}

// code after if
```

```
int number = 10;

if (number < 0) {
  // code
}

// code after if
```

Working of Java if statement



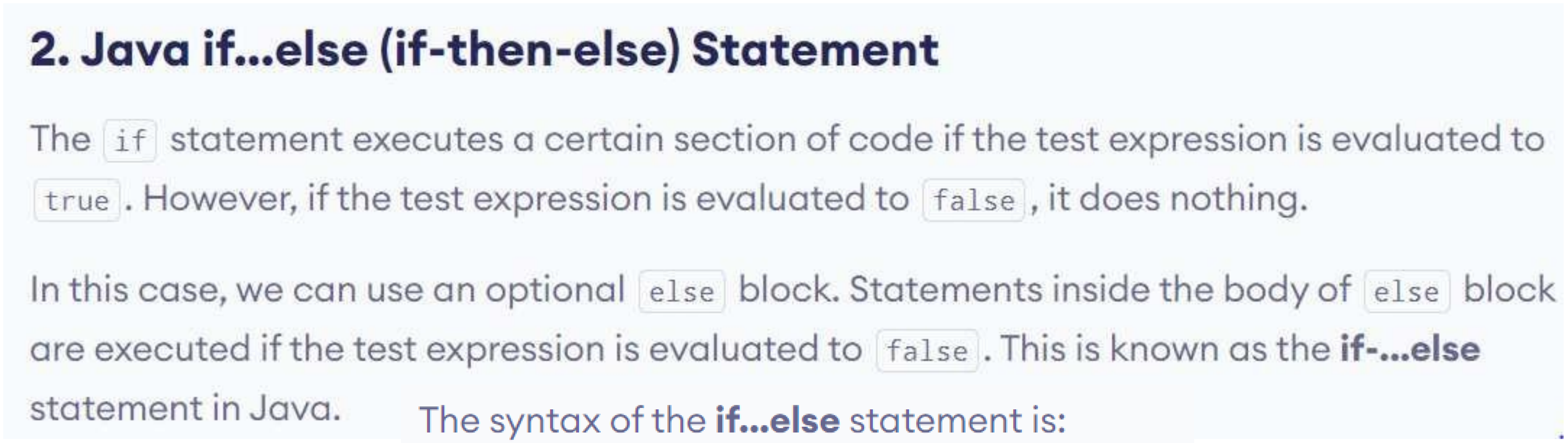
Example 1: Java if Statement

```
class IfStatement {
  public static void main(String[] args) {

    int number = 10;

    // checks if number is greater than 0
    if (number > 0) {
      System.out.println("The number is positive.");
    }

    System.out.println("Statement outside if block");
  }
}
```

Output

```
The number is positive.
Statement outside if block
```

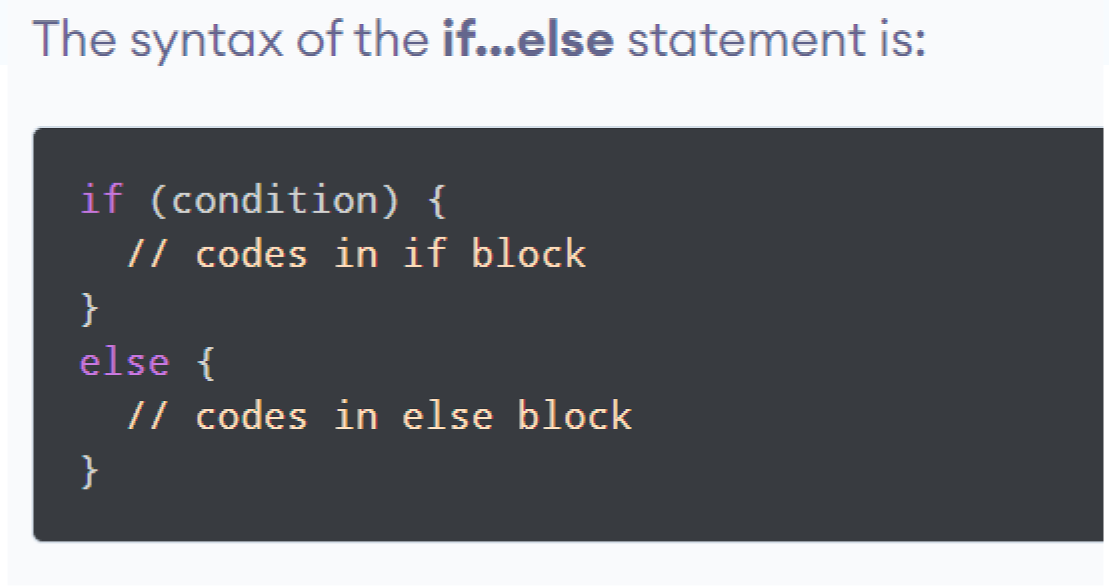
if…else Statement



2. Java if...else (if-then-else) Statement

The `if` statement executes a certain section of code if the test expression is evaluated to `true`. However, if the test expression is evaluated to `false`, it does nothing.

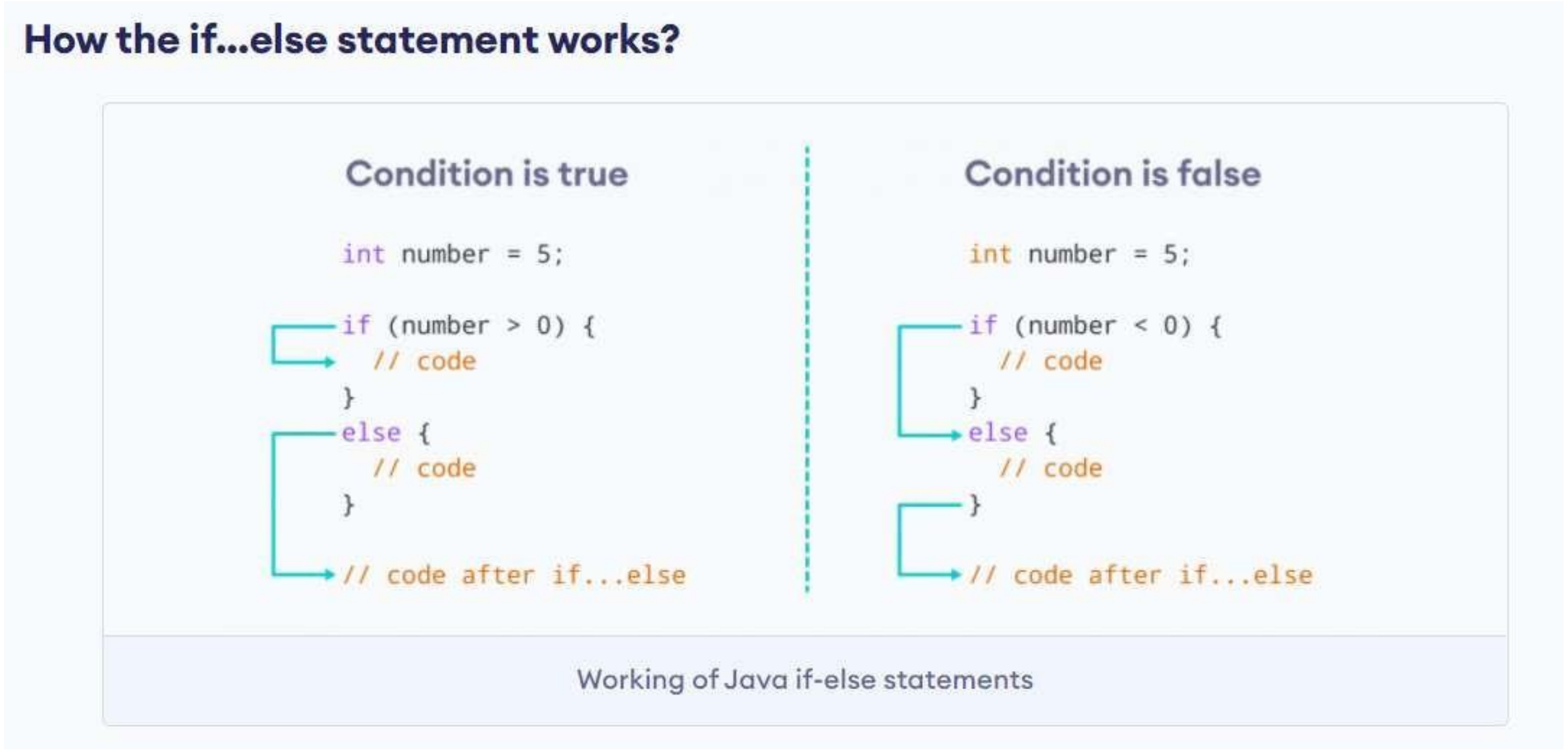
In this case, we can use an optional `else` block. Statements inside the body of `else` block are executed if the test expression is evaluated to `false`. This is known as the **if-...else** statement in Java.



The syntax of the **if...else** statement is:

```
if (condition) {
  // codes in if block
}
else {
  // codes in else block
}
```

if…else Statement



How the if...else statement works?

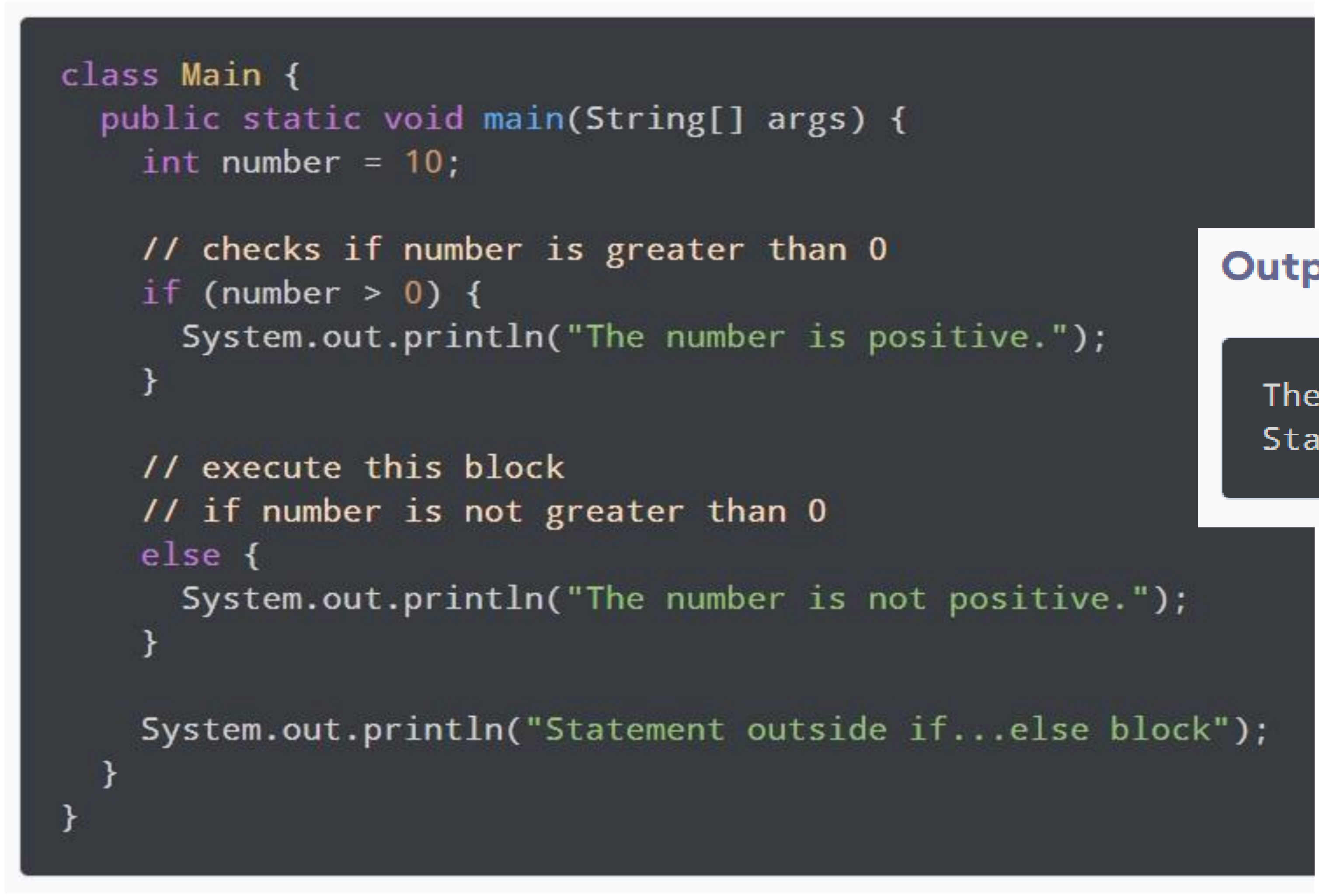
```
int number = 5;

if (number > 0) {
  // code
}
else {
  // code
}

// code after if...else
```

```
int number = 5;

if (number < 0) {
  // code
}
else {
  // code
}

// code after if...else
```

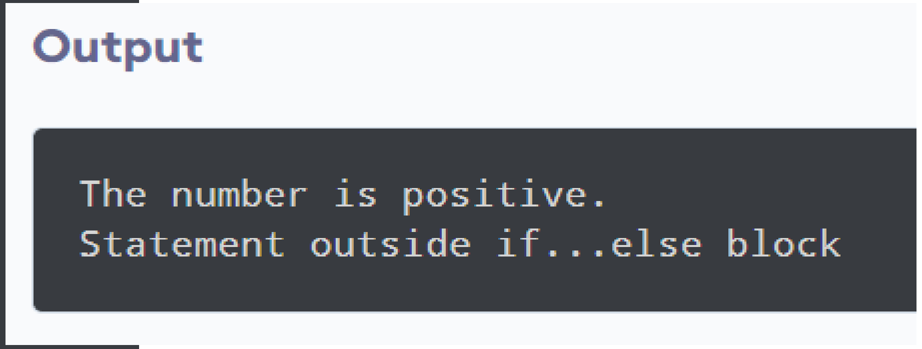
Working of Java if-else statements

if…else Statement

Example **Java if...else Statement**



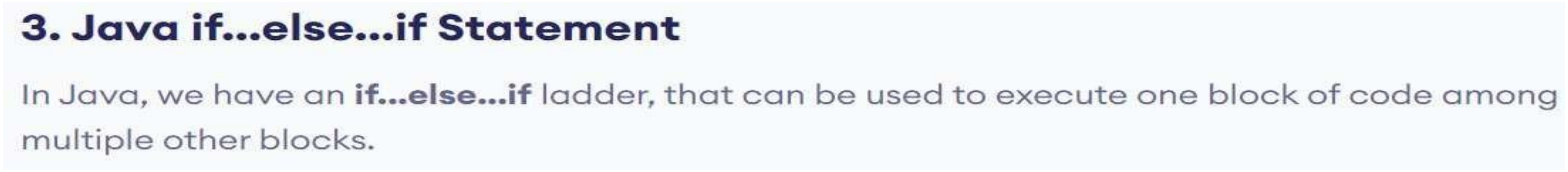
```
class Main {
  public static void main(String[] args) {
    int number = 10;

    // checks if number is greater than 0
    if (number > 0) {
      System.out.println("The number is positive.");
    }

    // execute this block
    // if number is not greater than 0
    else {
      System.out.println("The number is not positive.");
    }

    System.out.println("Statement outside if...else block");
  }
}
```



Output

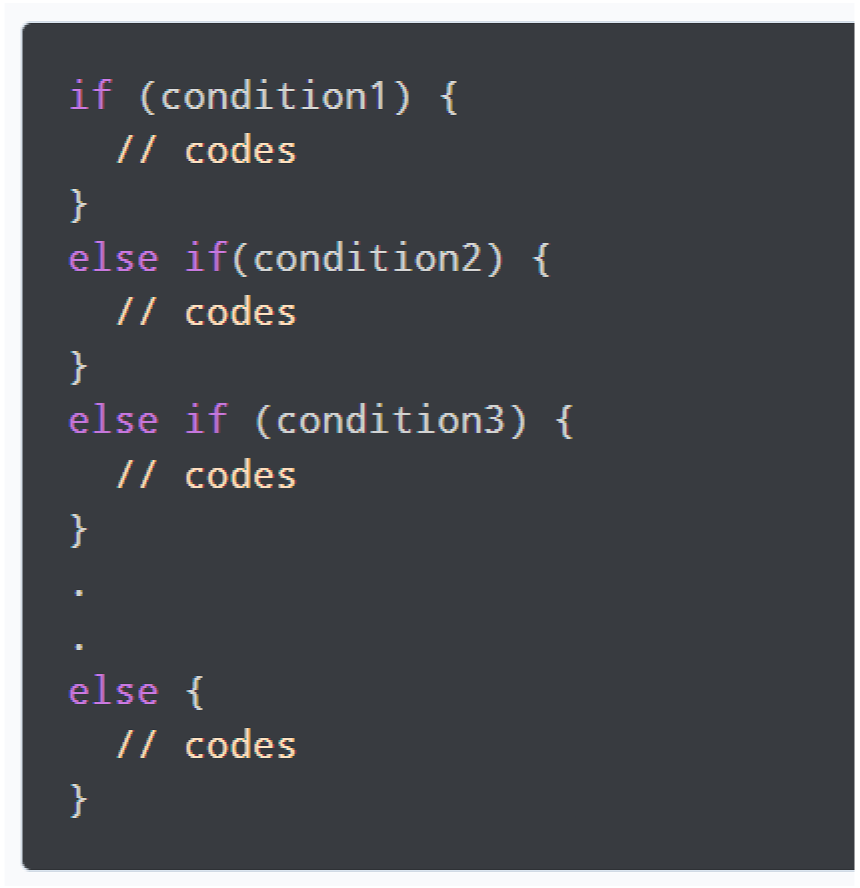
```
The number is positive.
Statement outside if...else block
```

if…else…if Statement



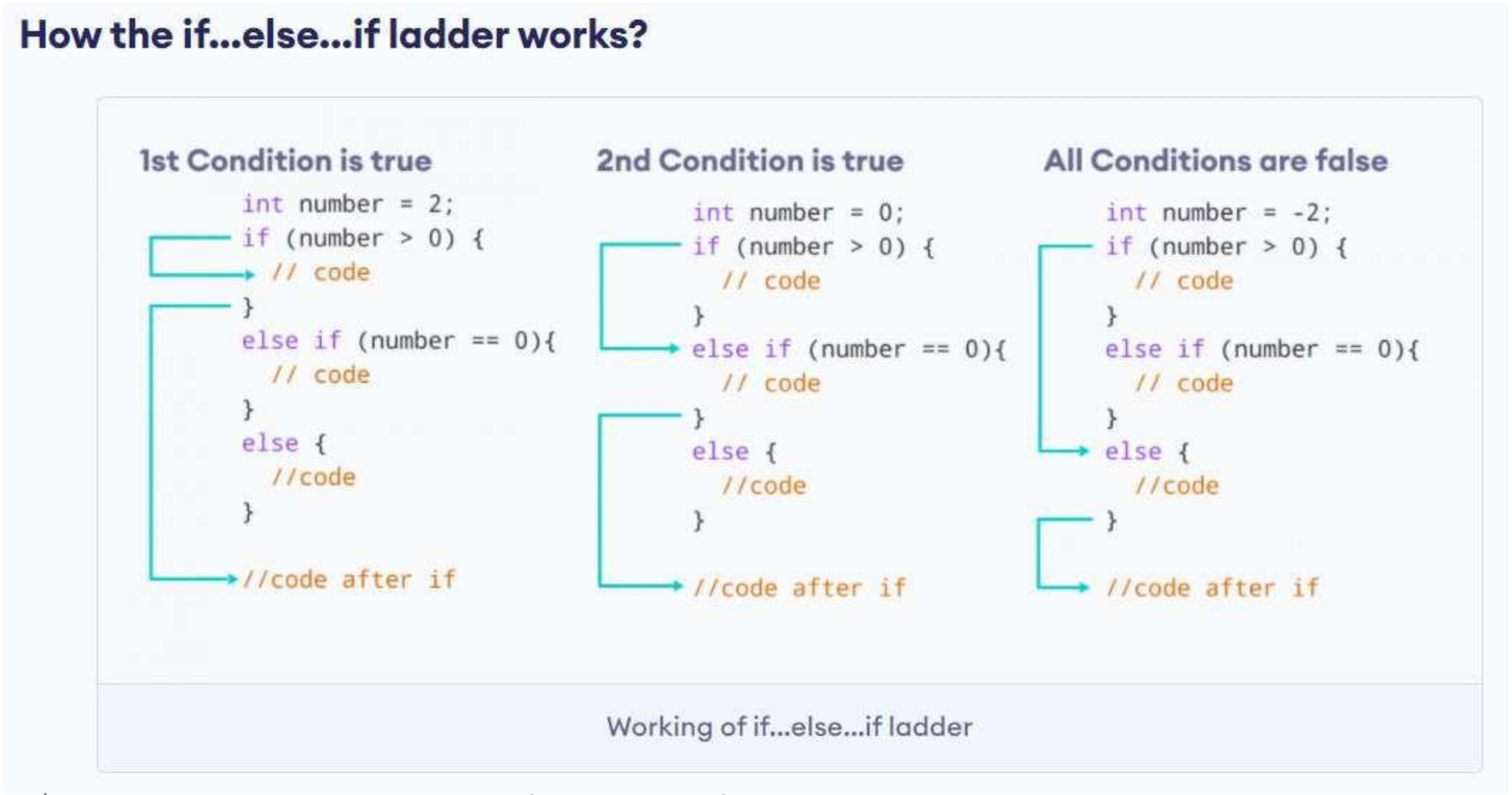
3. Java if...else...if Statement

In Java, we have an **if...else...if** ladder, that can be used to execute one block of code among multiple other blocks.



```
if (condition1) {
  // codes
}
else if(condition2) {
  // codes
}
else if (condition3) {
  // codes
}
.
.
else {
  // codes
}
```

if…else…if Statement



How the if...else...if ladder works?

1st Condition is true

```
int number = 2;
if (number > 0) {
  // code
}
else if (number == 0){
  // code
}
else {
  //code
}

//code after if
```

2nd Condition is true

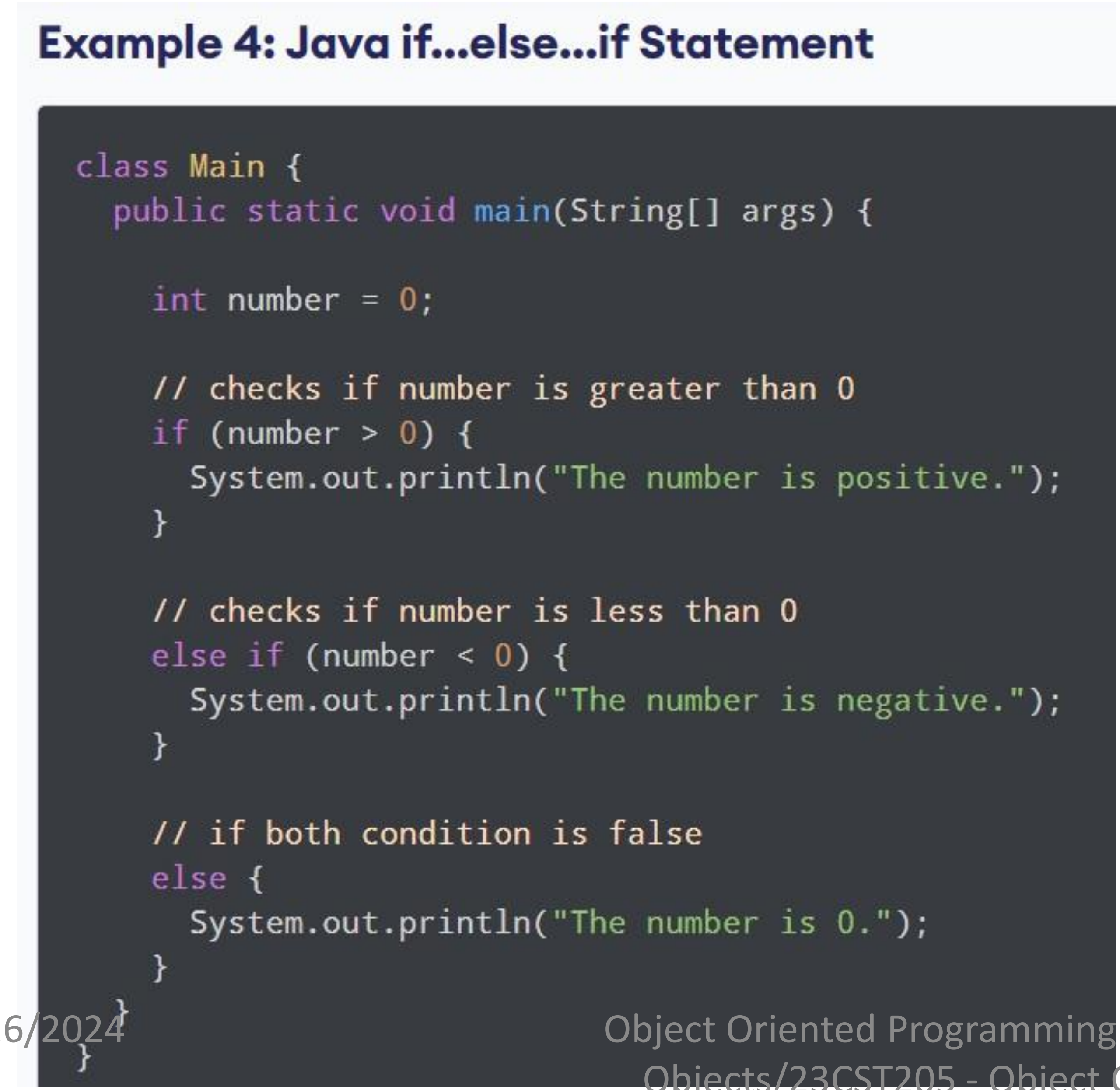
```
int number = 0;
if (number > 0) {
  // code
}
else if (number == 0){
  // code
}
else {
  //code
}

//code after if
```

All Conditions are false

```
int number = -2;
if (number > 0) {
  // code
}
else if (number == 0){
  // code
}
else {
  //code
}

//code after if
```

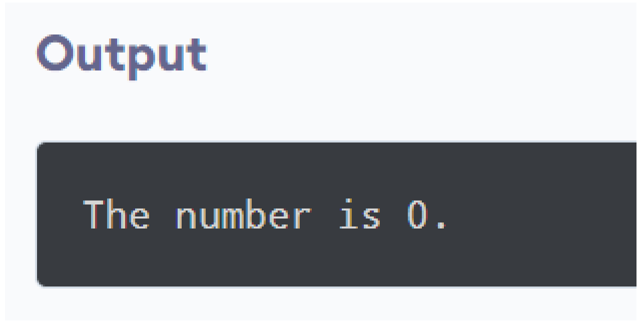
Working of if...else...if ladder

if…else…if Statement



Example 4: Java if...else...if Statement

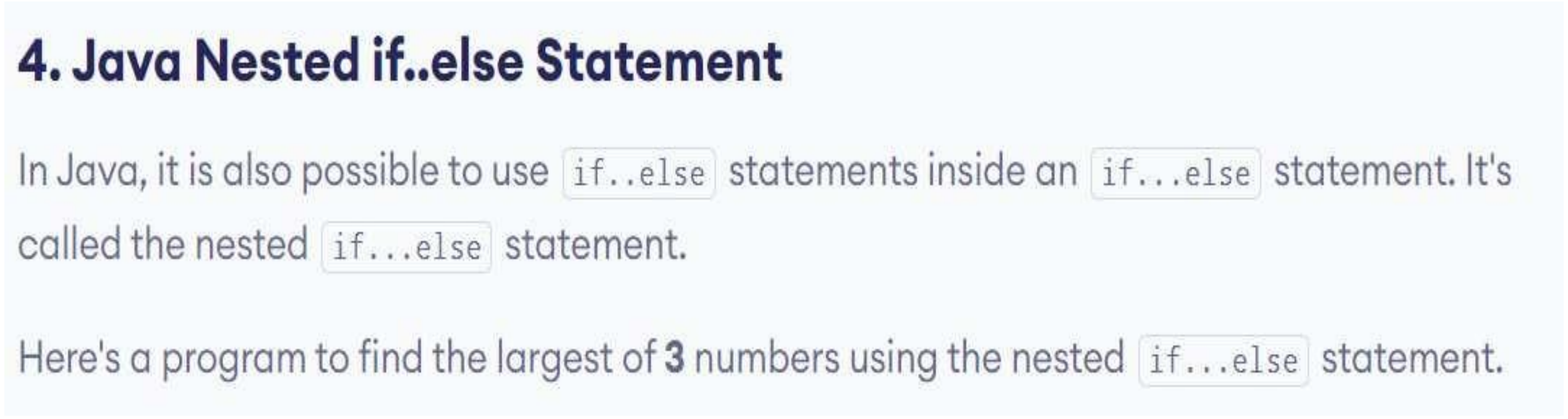
```
class Main {
  public static void main(String[] args) {

    int number = 0;

    // checks if number is greater than 0
    if (number > 0) {
      System.out.println("The number is positive.");
    }

    // checks if number is less than 0
    else if (number < 0) {
      System.out.println("The number is negative.");
    }

    // if both condition is false
    else {
      System.out.println("The number is 0.");
    }
  }
}
```



Output

```
The number is 0.
```

Nested if…else Statement

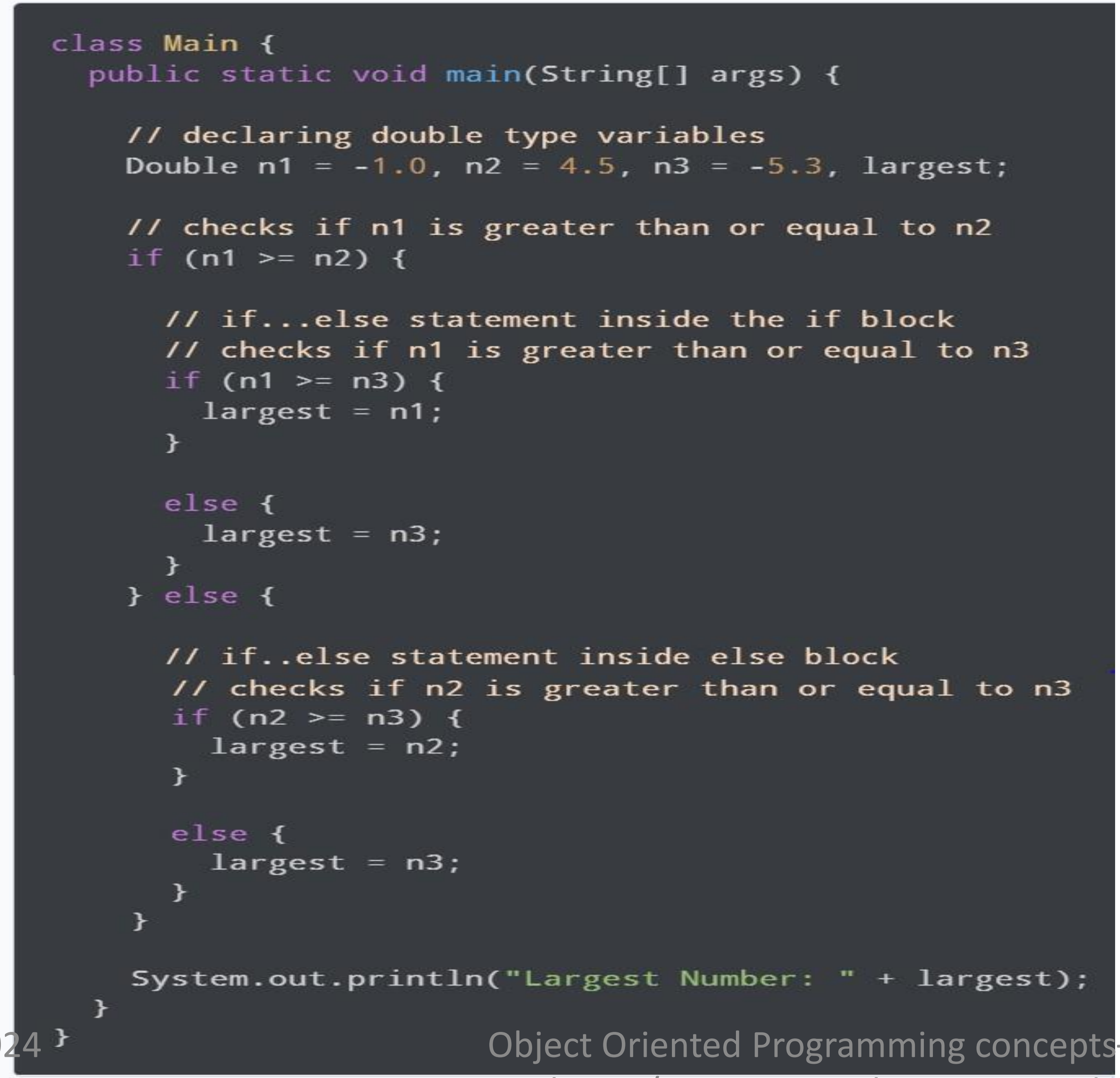


4. Java Nested if..else Statement

In Java, it is also possible to use `if..else` statements inside an `if...else` statement. It's called the nested `if...else` statement.

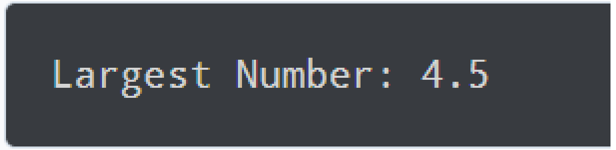
Here's a program to find the largest of **3** numbers using the nested `if...else` statement.

Example 5: Nested if...else Statement



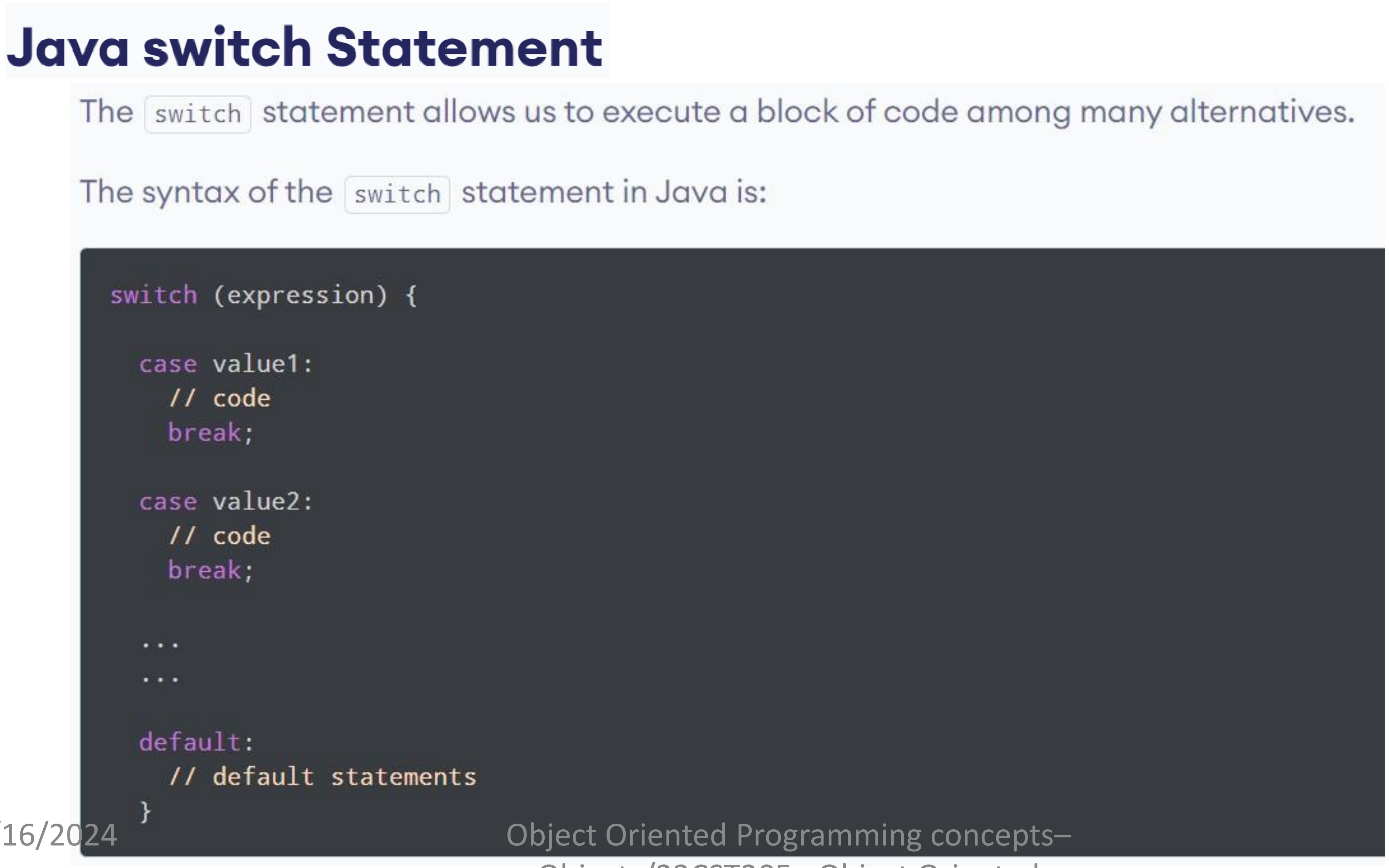
```
class Main {
  public static void main(String[] args) {

    // declaring double type variables
    Double n1 = -1.0, n2 = 4.5, n3 = -5.3, largest;

    // checks if n1 is greater than or equal to n2
    if (n1 >= n2) {

      // if...else statement inside the if block
      // checks if n1 is greater than or equal to n3
      if (n1 >= n3) {
        largest = n1;
      }

      else {
        largest = n3;
      }
    } else {

      // if..else statement inside else block
      // checks if n2 is greater than or equal to n3
      if (n2 >= n3) {
        largest = n2;
      }

      else {
        largest = n3;
      }
    }

    System.out.println("Largest Number: " + largest);
  }
}
```

Output:



```
Largest Number: 4.5
```

Java switch Statement



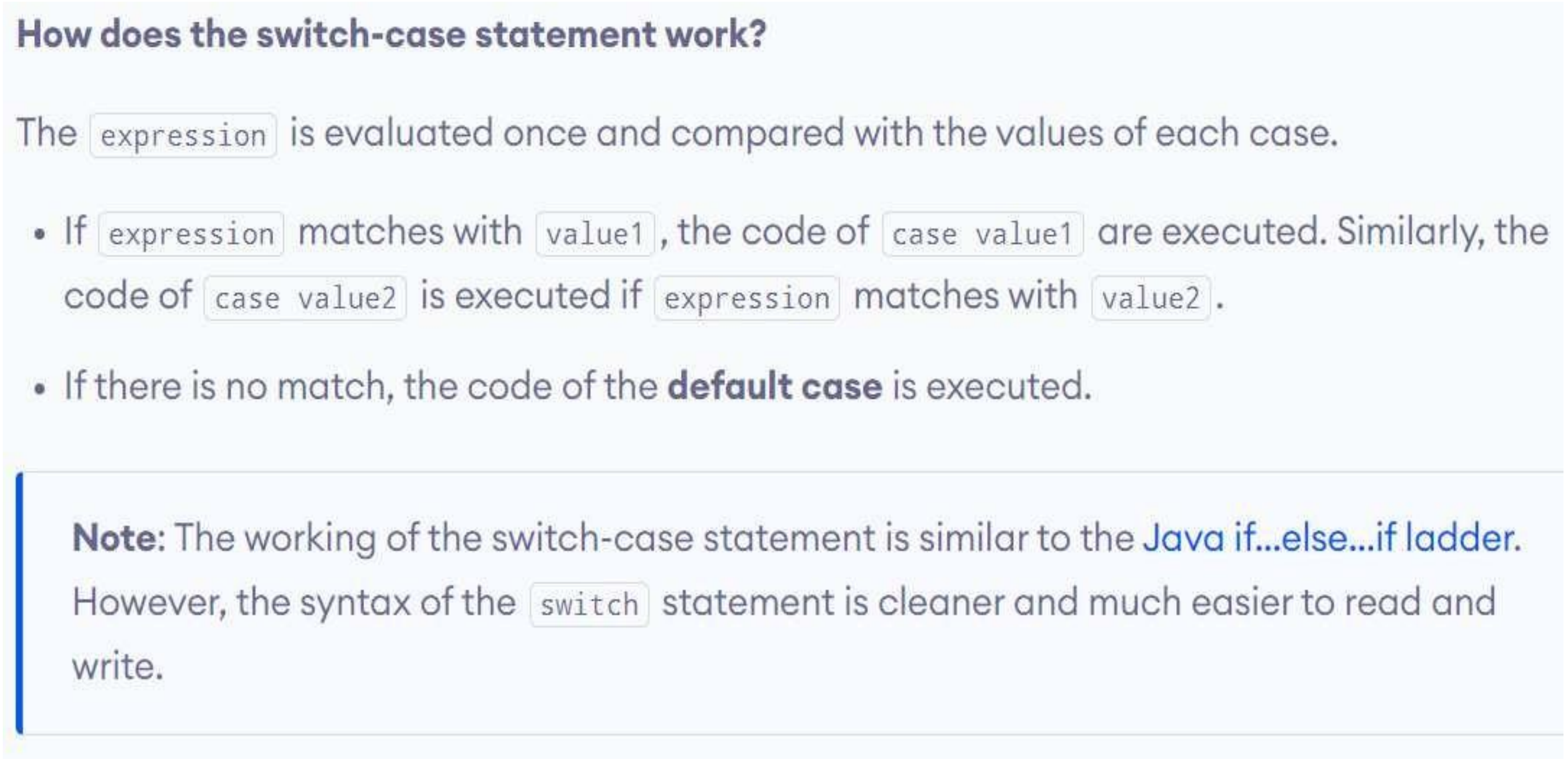
Java switch Statement

The `switch` statement allows us to execute a block of code among many alternatives.

The syntax of the `switch` statement in Java is:

```
switch (expression) {

  case value1:
    // code
    break;

  case value2:
    // code
    break;

  ...
  ...

  default:
    // default statements
  }
```

Java switch Statement



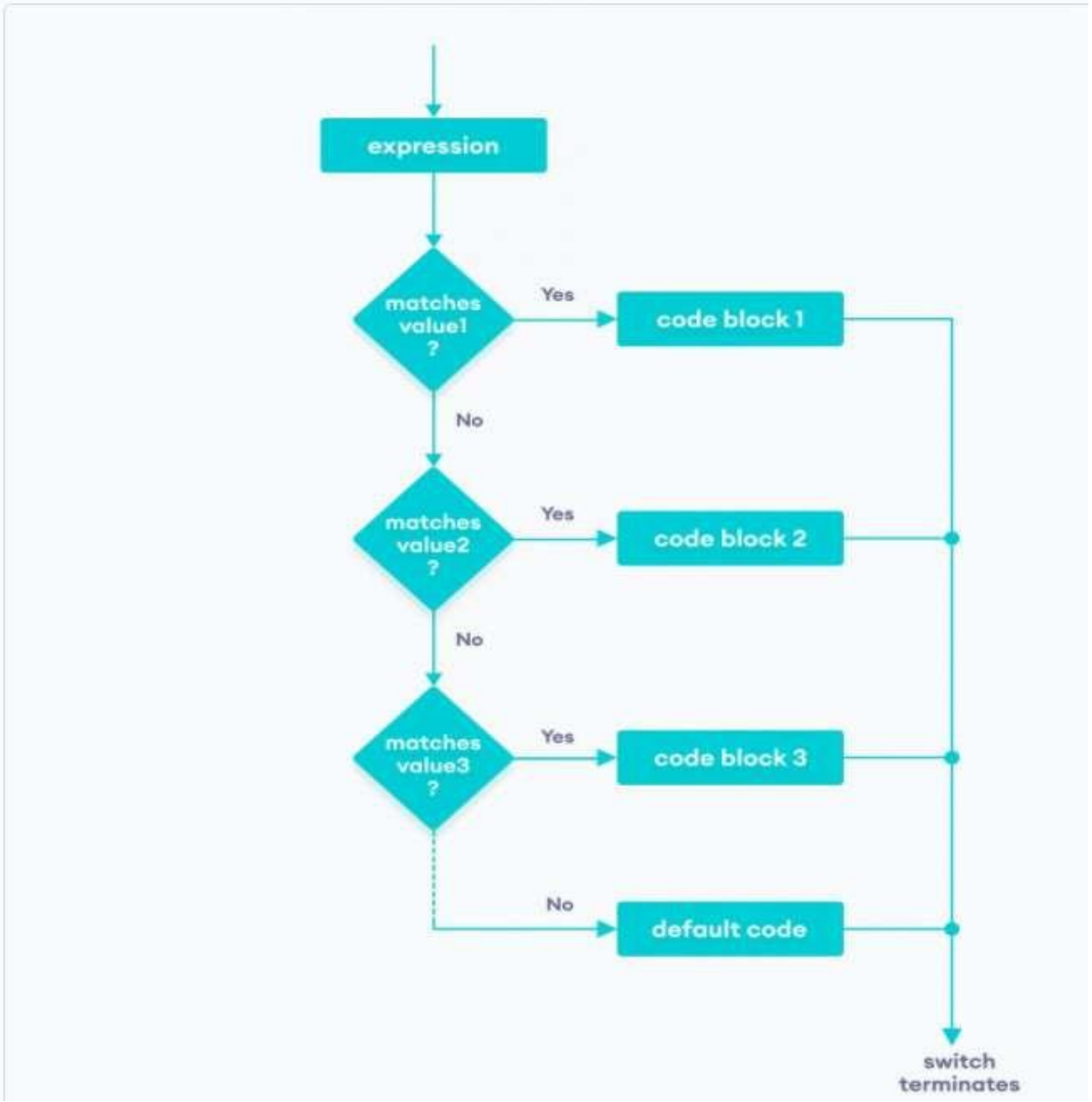
How does the switch-case statement work?

The `expression` is evaluated once and compared with the values of each case.

- If `expression` matches with `value1`, the code of `case value1` are executed. Similarly, the code of `case value2` is executed if `expression` matches with `value2`.
- If there is no match, the code of the **default case** is executed.

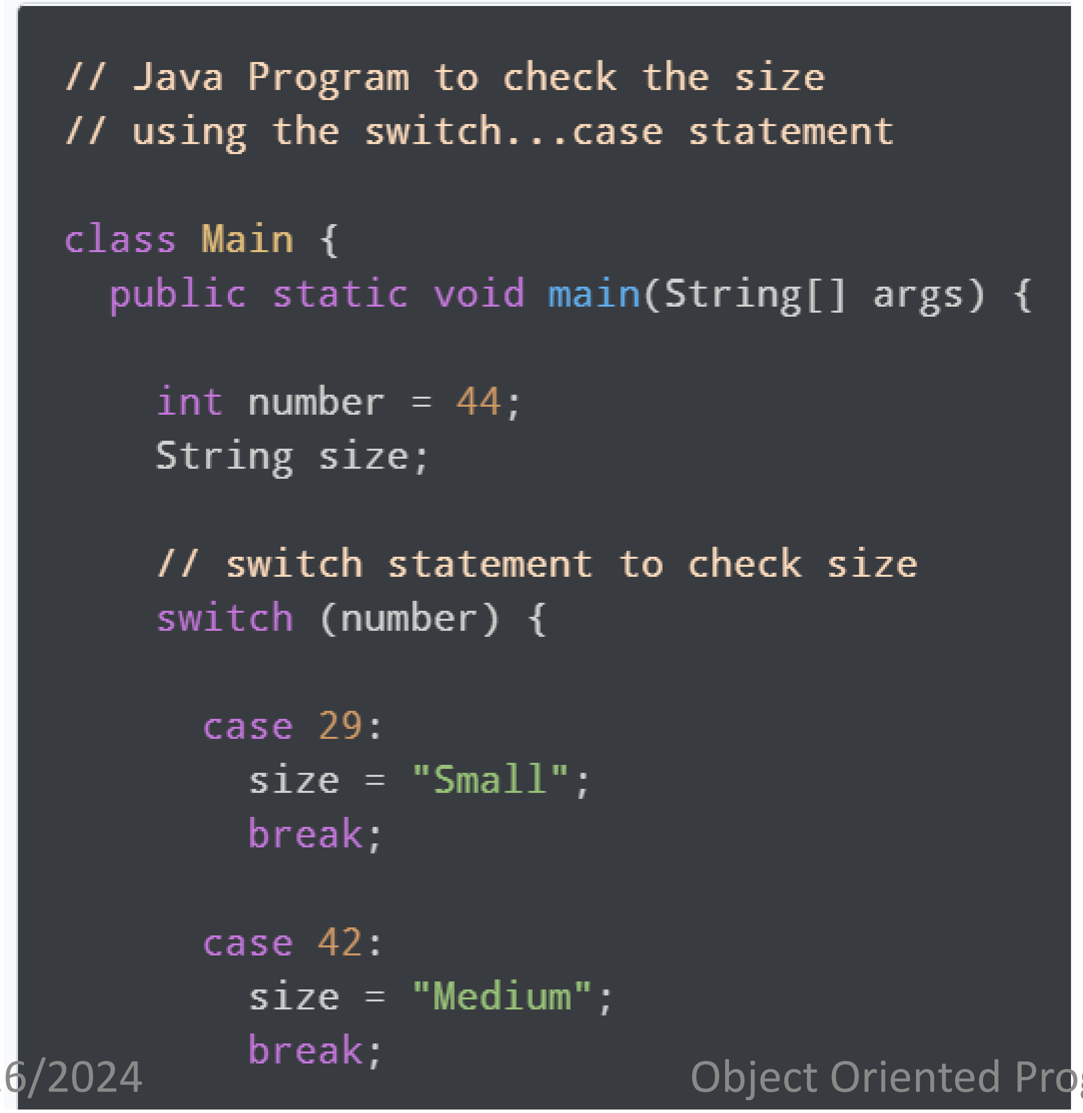
Note: The working of the switch-case statement is similar to the Java if...else...if ladder. However, the syntax of the `switch` statement is cleaner and much easier to read and write.

Flowchart of switch Statement

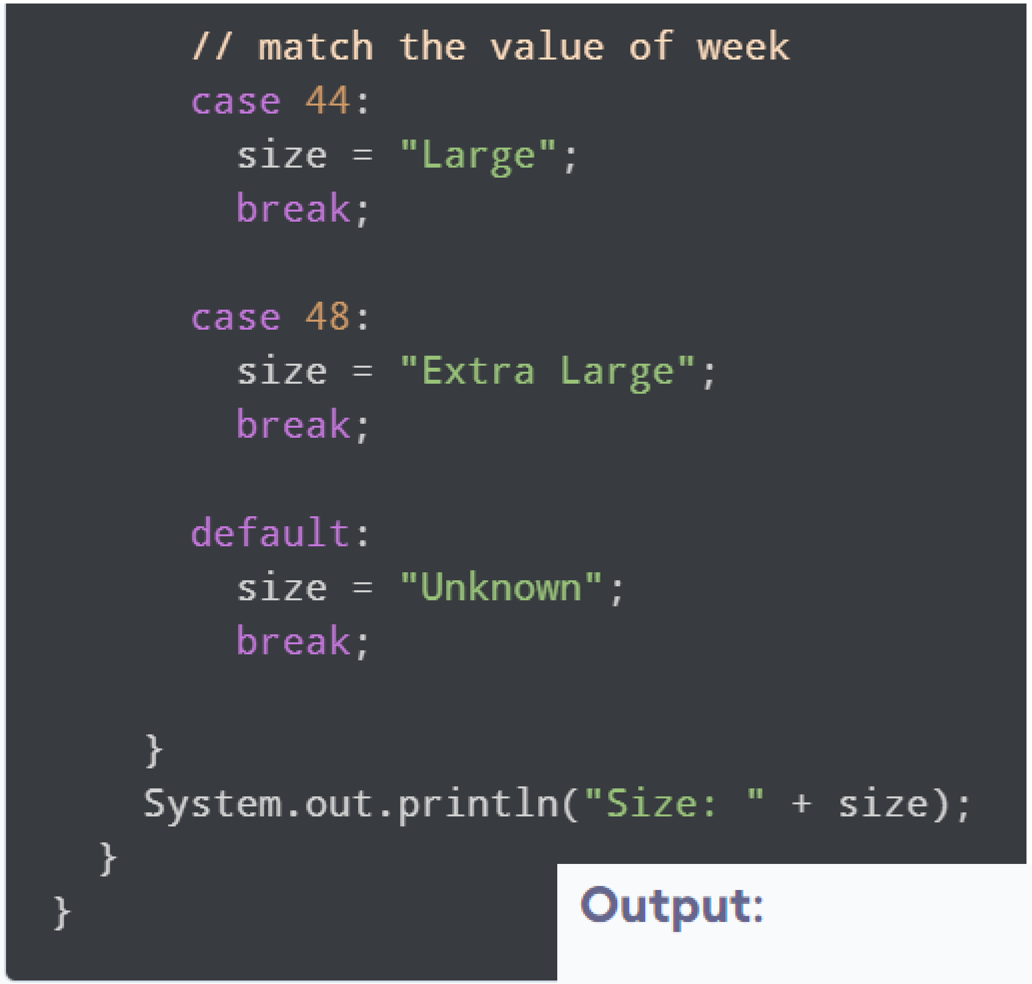


Java switch Statement

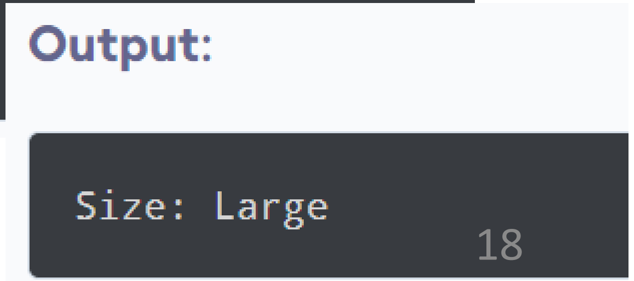
Example: Java switch Statement



```
// Java Program to check the size
// using the switch...case statement

class Main {
  public static void main(String[] args) {

    int number = 44;
    String size;

    // switch statement to check size
    switch (number) {

      case 29:
        size = "Small";
        break;

      case 42:
        size = "Medium";
        break;
```



```
      // match the value of week
      case 44:
        size = "Large";
        break;

      case 48:
        size = "Extra Large";
        break;

      default:
        size = "Unknown";
        break;

    }
    System.out.println("Size: " + size);
  }
}
```



Output:

```
Size: Large
```

THANK
YOU!